Report: Implementation of OAuth2 Login with Google

Objective
Pandora Company Limited required a secure PHP-based login system integrating "Login with Google" functionality using OAuth2 to streamline authentication.

Implementation Process

1. Setting Up Google Cloud Platform

- Project Creation: Created *Pandora Authentication* project in Google Cloud Console.
- OAuth Consent Screen: Configured application name, scopes (email and profile), and redirect URI (http://localhost/pandora/redirect.php).
- OAuth Credentials: Generated Client ID and Secret, matching the redirect URI.

2. PHP Application Development

Developed three primary files:

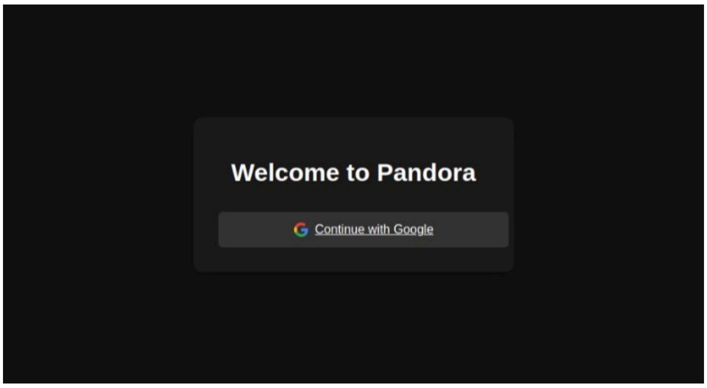
- login.php: Displays a "Login with Google" button and initializes the Google PHP client.
- redirect.php: Handles the redirect, fetches an access token, and stores it in a session.
- dashboard.php: Verifies the session token, fetches user details, and displays a greeting.

Challenges and Solutions

1. Missing Scopes:
    - *Problem*: Forgot to include required scopes in OAuth consent screen.
    - *Solution*: Updated scopes and verified permissions.
2. Authorization Error in Testing Mode:
    - *Problem*: Testing mode blocked unauthorized accounts.
    - *Solution*: Added test accounts; later pushed to Production mode.
3. Logo Verification:
    - *Problem*: Adding a logo required app verification.
    - *Solution*: Temporarily removed logo to avoid delays.
4. Session Management:
    - *Problem*: Sessions did not persist across pages.
    - *Solution*: Ensured session_start() was included in all files.
5. Google PHP Client Library Setup:
    - *Problem*: Missing dependencies caused library failure.
    - *Solution*: Installed the library using Composer and verified autoloading.
6. Access Token Exchange:
    - *Problem*: Mismatch between redirect URIs.
    - *Solution*: Updated URIs in both Google Cloud Console and Google_Client

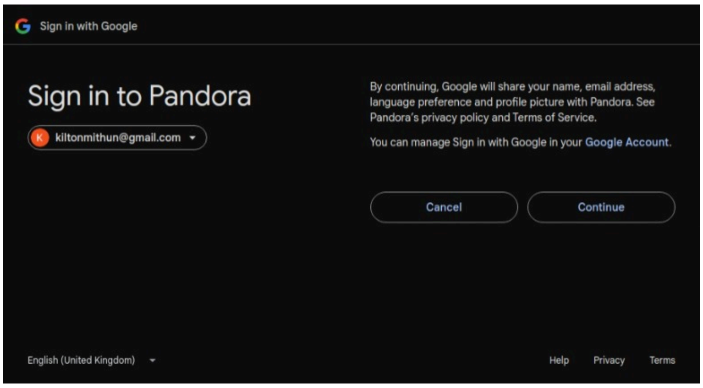
Screenshots

1. Login Page (login.php):
   Displays a "Login with Google" button



Welcome to Pandora

2. Authentication Flow:
   Redirects to Google's login page and requests permissions for email and profile.



Sign in with Google

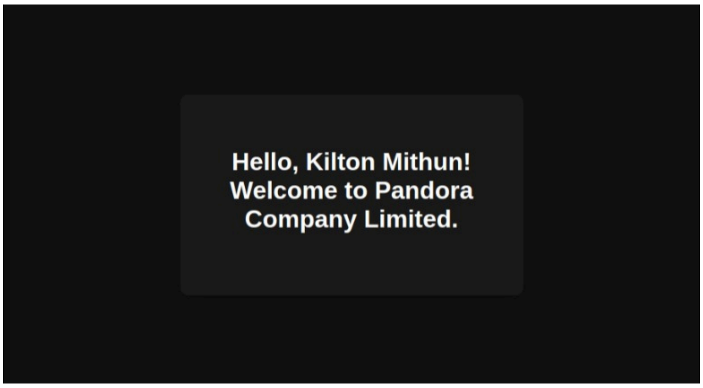
Sign in to Pandora

kiltonmithun@gmail.com

By continuing, Google will share your name, email address, language preference and profile picture with Pandora. See Pandora's privacy policy and Terms of Service.

You can manage Sign in with Google in your Google Account.

3. Dashboard (dashboard.php):
   Displays a personalized message



Hello, Kilton Mithun!
Welcome to Pandora
Company Limited.

Conclusion

This implementation successfully integrates OAuth2-based "Login with Google" functionality into Pandora Company Limited’s PHP application. By leveraging the Google Identity Services API, the solution ensures secure and streamlined user authentication.

References

1. Setting up OAuth 2.0
2. Setting up your OAuth consent screen
3. Google PHP Client Library on GitHub
4. PHP Official Documentation on Session Management